



ALGORITMOS, DATOS y PROGRAMAS

RECURSANTES Modalidad Semipresencial

Año 2018

Carrera/ Plan:

Licenciatura en Informática

Plan 2007 y Plan 2012

Licenciatura en Sistemas

Plan 2007 y Plan 2012

Analista Programador Universitario

Plan 2007

Año: 1°

Régimen de Cursada: Anual

Carácter: Obligatoria

Correlativas: Expresión de Problemas y Algoritmos- Conceptos de Organización de Computadoras- Matemática 0

Coordinador: Armando De Giusti

Profesores: Cristina Madoz, Eduardo Ibañez.

Hs. semanales: 7 (teoría y práctica)

FUNDAMENTACIÓN

Se trabaja a partir de diferentes situaciones problemáticas de la vida real y se abordan las estrategias de solución bajo criterios de calidad, eficiencia y corrección, para finalmente llegar a una implementación acorde a las especificaciones planteadas.

Los conceptos abordados en la asignatura permitirán al alumno familiarizarse con los conceptos básicos de la disciplina.

Objetivos Generales:

- Analizar problemas resolubles con computadora, poniendo énfasis en la modelización, abstracción de funciones y en la descomposición funcional de los mismos. Obtener una expresión sintética y precisa de los problemas, con una documentación de una metodología de trabajo por el alumno.
- Estudio, expresión simbólica, implementación y evaluación de algoritmos, orientando los mismos a la resolución de las partes (módulos) en que se descomponen los problemas, a partir de un paradigma procedural/imperativo.
- Introducción de las nociones de estructuras de datos, tipos de datos y abstracción de datos.
- Introducción de los conceptos de corrección y eficiencia de algoritmos.
- Introducción de los conceptos básicos de un segundo paradigma de programación (orientación a objetos) con énfasis en la noción de reusabilidad.
- Combinar los elementos mencionados anteriormente a fin de que el alumno complete el ciclo del problema a su solución con computadora, analizando simultáneamente algoritmos y datos.
- Introducción de los conceptos básicos de la Programación Concurrente



Contenidos Mínimos

- Modelización de problemas del mundo real.
- Algorítmica. Estructuras de control.
- Tipos de datos simples y compuestos. (estáticos y dinámicos).
- Procedimientos y funciones. Concepto de reusabilidad.
- Estructuras de datos no lineales.
- Recursividad.
- Tipos de datos abstractos.
- Estrategias de diseño de algoritmos.
- Eficiencia, legibilidad y depuración de algoritmos
- Conceptos básicos de Programación Orientada a Objetos y Programación Concurrente.

Programa

A- Introducción. Conceptos básicos

Definiciones.

Modelización de problemas del mundo real.

Del problema real a su solución por computadora.

Software.

B - Algoritmos. Acciones elementales

Estructuras de control. Modelo de máquina abstracta.

El robot como máquina abstracta. Operaciones elementales del robot.

Estructura esquemática de un programa para una máquina abstracta (robot).

Corrección de algoritmos. Importancia de la verificación.

Eficiencia de un algoritmo.

Importancia de la documentación de un algoritmo.

Relación de los conceptos anteriores con el modelo de máquina abstracta.

Ejemplos.

C –Datos y Tipos de datos

Constantes y variables.

Tipos de datos simples y compuestos.

Tipos de datos primitivos.

Tipos de datos definidos por el usuario.

Funciones predefinidas.

Tipos ordinales.

D- Modularización. Procedimientos y funciones. Parámetros

Descomposición de problemas. Utilidad e importancia.

La noción de reusabilidad.

Subprogramas o módulos.

Procedimientos.

Funciones.

Conceptos de argumentos y parámetros.

Conceptos de variables locales y variables globales.

Procedimientos y funciones con parámetros.



Manejo de memoria en ejecución.

E- Estructuras de datos

Introducción y clasificación de las estructuras de datos.

Registros.

Arreglos.

Arreglos. Operaciones con arreglos de una dimensión.

Matrices. Tratamiento de información estructurada en vectores y matrices.

Algoritmos de búsqueda.

Algoritmos de ordenación. Ordenación por índice.

F- Estructura de datos enlazadas: listas.

Alocación dinámica. Punteros.

Listas. Operaciones con listas.

Listas doblemente enlazadas y circulares. Características y operaciones.

G- Recursividad

Características.

Ejecución de un programa y la pila de activación. Manejo de memoria en ejecución.

Análisis comparativo entre soluciones iterativas y recursivas.

Ejemplos.

H- Concepto de Corrección. Análisis de algoritmos: concepto de eficiencia

Concepto de corrección. Técnicas para medir corrección. Verificación de Programas.

Concepto de eficiencia.

Análisis de eficiencia de un algoritmo.

Análisis de algoritmos según su tiempo de ejecución y su utilización de memoria.

Análisis de Algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Notación $O()$.

Análisis de eficiencia en algoritmos recursivos.

Análisis de eficiencia en algoritmos de búsqueda y ordenación sobre vectores.

Métodos de ordenación eficientes.

Algoritmos numéricos y propagación de error.

J- Estructura de datos no lineales: árboles.

Introducción al concepto de datos no lineales.

Terminología y definiciones básicas del tipo de dato árbol.

Arboles binarios. Representación y operaciones.

Arboles binarios ordenados. Representación y operaciones.

Problemas que combinen árboles, listas y arreglos.

K- Tipos de datos abstractos

Abstracción de datos.

Conceptos sobre tipos de datos.

Módulos, interfaz e implementación. Encapsulamiento de datos.



Diferencia entre tipo de dato y tipo abstracto de dato.
Requerimientos y diseño de TADs. Ejemplos TAD pila, TAD cola.

L- Introducción a la Programación Orientada a Objetos

Motivación. Reusabilidad de soluciones.
Abstracción de datos y procesos.
La noción de Objeto. Operaciones (métodos) aplicables a un objeto.
Concepto de clases e instancias.
Noción de herencia. Relación con el re-uso.
Aplicaciones.
Características de los lenguajes enfocados a POO.

M - Conceptos iniciales de concurrencia

Motivación: arquitecturas de computadoras actuales, aprovechamiento de los procesadores.
Definiciones.
Ejemplos.

Bibliografía Básica

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Estructuras de Datos y Algoritmos.

Hernández R., Dormido R., Lazaro J. Ros S. Pearson Education. 2000.

Introduction to algorithms

Comen, Leiserson. MIT Press 2001.

Estructuras de Datos y Algoritmos.

Aho Alfred, Hopcroft John y Ullman Jeffrey. Addison Wesley Publishing Company. EUA. 1998.

Programación en Pascal

Joyanes Aguilar, Luis. Mc Graw Hill. 2006

Fundamentos de Programación. Libro de Problemas.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 2003.

Data structures, algorithms and software principles.

Standish, T. A. Addison Wesley Publishing Company. 1994.

Estructuras de Datos y Algoritmos

Weiss, M.A. Addison Wesley. 1995.

Fundamentos de Programación.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Algoritmos y estructuras de datos y programación orientada a objetos.

Flórez Rueda. Ecoe Ediciones. Bogotá. 2005. ISBN 958648394/0

Programación En C Metodología, Algoritmos Y Estructura De Datos.

Joyanes Aguilar Luis – Zahonero Martínez. Segunda Edición –Editorial Mc Graw Hill. España - Edición 2007



Bibliografía Adicional

Introduction to Computer Science with applications in Pascal.

Garland, S.J. Addison Wesley Publishing Company. 1986.

Estructuras de Datos.

Franch Gutierrez, Xavier. Alfaomega Grupo Editor Argentino.2002

Estructura de Datos.

Joyanes Aguilar C., Zahonero Martinez I. Mc Graw Hill. 1998.

Estructuras de Datos. Libro de Problemas.

Joyanes Aguilar L., Fernandez M., Rodriguez L. Mc Graw Hill. 1999.

Estructuras de Datos.

Lipschutz, S. Mc Graw Hill. 1997.

Programación estructurada en Turbo Pascal 7.

Lopez Roman, L. Alfaomega Grupo Editor Argentino. 1998.

Estructuras de Datos.

Martinez Román, Quiroga Elda. Thomson International. 2002

Estructura de Datos y Algoritmos.

Sisa, Alberto Jaime. Editorial Prentice. 2002.

Pascal Estructurado.

Tremblay, Jean Paul. Mc Graw Hill.1980.

Data structures, algorithms and performance.

Wood, D. Addison Wesley Publishing Company. 1993.

Structures and Algorithm Analysis in Java

Weiss, M.A. Data, 3rd Edition, Pearson/Addison Wesley, 2011

Data Structures and Algorithms using C#.

M. McMillan. Cambridge University Press, 2006

Sitios de interés:

<http://csunplugged.org>

<http://www.eduteka.org>



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

2. Metodología de Enseñanza

A los fines de la organización de la cátedra se propone una modalidad semipresencial y los alumnos que pueden cursar son aquellos que no hayan aprobado la cursada de la asignatura ADP.

La cátedra está conformada por un Profesor, un Jefe de Trabajos Prácticos y Auxiliares Docentes.

Los alumnos tendrán Auxiliares Docentes designados a los efectos de responder preguntas y consultas sobre las guías de práctica y entrega de las actividades.

Respecto al Material teórico utilizado está disponible en el EVEA IDEAS que utiliza la cátedra dado que no es obligatoria la asistencia a las clases de teoría. Sin embargo, se recomienda al alumno asistir a cualquiera de los turnos presenciales de CADP disponibles para resolver dudas y/o consultas correspondientes a los contenidos vistos durante el 1er cuatrimestre. Para los contenidos vinculados con el 2do cuatrimestre, se plantean encuentros presenciales en los cuales el Profesor y / o Jefe de Trabajos Prácticos atienden la dudas de los contenidos teóricos y con el Auxiliar Docente dicta o atiende las consultas de las prácticas o para la entrega de las actividades de los temas que consideran críticos para su comprensión.

Respecto a los contenidos prácticos, el alumno debe realizar y entregar un conjunto de actividades pautadas. Cada actividad tiene un plazo límite de entrega. La entrega y aprobación de las actividades requeridas permitirá al alumno acceder a rendir el examen parcial de la asignatura. Cada actividad será considerada como: A (Aprobada), D(Desaprobada) y N (No entregada). No se requiere que el alumno asista a las clases prácticas, pero se le recomienda asistir a las clases prácticas de los turnos presenciales de CADP para resolver dudas y/o consultas del 1er cuatrimestre y consultas a través del entorno para los temas correspondientes al 2do cuatrimestre.

Evaluación

Parcial de Trabajos Prácticos

- El curso práctico se evalúa mediante dos parciales prácticos.
- La aprobación de la cursada de la asignatura se obtiene con la aprobación de los dos parciales prácticos.
- El alumno dispondrá de 6 fechas de evaluación, debiendo tener aprobado el primer parcial para poder presentarse a rendir el segundo parcial para obtener la cursada de la asignatura.
- Antes de la fecha del parcial se publicarán las listas de los alumnos en condiciones de rendir, para que en caso de error, el alumno pueda plantear su situación al Jefe de su turno sólo en el horario que se establezca. Los alumnos que no estén en las listas publicadas para rendir los parciales NO podrán rendir la prueba correspondiente. Los Jefes de Trabajos Prácticos NO atenderán inconvenientes administrativos en los momentos previos a un parcial.
- Los parciales deben entregarse escritos en forma legible.
- Los alumnos deben presentarse a rendir con documento de identidad.

Muestra de Parciales

- Los parciales son corregidos por los Auxiliares Docentes y revisados por los jefes de trabajos prácticos.
- Los parciales pueden ser vistos y consultados por los alumnos en forma individual, personal y exclusivamente en el día y hora que se publique. Una vez que el alumno recibe el parcial NO puede retirarlo del aula y debe devolverlo al finalizar su revisión.
- Los resultados se publicarán en el EVEA IDEAS y en la cartelera de primer año que se encuentra en Planta Baja.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

CRONOGRAMA DE CLASES Y EVALUACIONES

Inicio: semana del 26/3

Semana	Contenidos	Actividades
Semana 1	Conceptos básicos – Datos y Tipos de datos	Práctica 1 – Repaso Ingreso (Robot)
Semana 2	Datos y Tipos de datos (cont.) Estructuras de Control	Practica 2 – Máximos y mínimos (Pascal)
Semana 3	Tipos de datos definidos por el usuario. Modularización.	Práctica 3 – Estructuras de control (manejo de caracteres + procesamiento de números). Tipo subrango y string.
Semana 4	Modularización (cont.)	Práctica 4 – Funciones y Procedimientos I Práctica 5 – Funciones y Procedimientos II
	Encuentro, consultas para posterior entrega de la actividad 1.	
Semana 5	Estructuras de datos. Registros	Práctica 5 (cont.) Práctica 6 – Registros
Semana 6	Arreglos: conceptos y operaciones con vectores.	Práctica 6 (cont.)
Semana 7	Arreglos (cont.) Métodos de búsqueda en vectores.	Práctica 7 – Arreglos I
Semana 8	Punteros. Listas	Práctica 8 – Arreglos II
	Encuentro, consulta para posterior entrega de la actividad 2.	
Semana 9	Listas	Práctica 9 – Punteros Práctica 10 - Listas
Semana 10	Corrección. Eficiencia	Práctica 10 (cont.)
Semana 11	Eficiencia	Práctica 11 - Repaso
	Encuentro, consultas para posterior entrega de la actividad 3.	
Semana 12	Recursión	Consulta
Semana 13		Consultas
Semana 14		Examen Parcial
Semana 15		Consultas y 1er Recuperatorio
Semana 16		Consultas y 2da Recuperatorio

Semana	Contenidos	Actividades
Semana 17	Recursión.	Práctica – Repaso de los contenidos prácticos del primer cuatrimestre.
Semana 18	Tipos de datos no lineales. Definición y características. Ejemplos. Tipo de Datos: Arboles	Practica – Recursión
Semana 19	Tipos de datos Arboles Binarios: Definición y características: Usos	Práctica – Arboles



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

	Encuentro presencial y consultas	
Semana 20	Tipos de datos Arboles Binarios Ordenados: Definición y características. Principales operaciones	Práctica – Arboles continuación.
	Entrega de la actividad 4	
Semana 21	Arboles Binarios ordenados continuación.	Práctica Continuación.
	Encuentro presencial y consultas	
Semana 22	Tipos de datos Abstractos : concepto y características.	Práctica Tipo de datos Abstractos.
Semana 23	Clasificación dentro del cuadro de tipos de datos y destacar sus ventajas y ejemplos.	Práctica – Continuación.
Semana 24	Tipos de datos abstractos: su implementación y usos.	Práctica Continuación
	Entrega de la Actividad 5	
Semana 25	TAD de TAD. Como usar un tipo de dato abstracto en un programa.	Consulta
Semana 26	Ejemplos.	Consulta
Semana 27	Consulta sobre el tema tipo de datos abstractos.	Consulta
	Encuentro y consultas. Posterior entrega de la actividad 6	
Semana 28	Nociones básica de Programación Orientada a Objetos	Consulta
Semana 29	Nociones básicas de programación Concurrente.	
Semana 30	Examen Parcial	
Semana 31	Consultas y Recuperatorio	
Semana 32	Consultas y Recuperatorio	

Contacto de la cátedra (mail, página, blog, plataforma virtual de gestión de cursos):

Contacto: progra@lidi.info.unlp.edu.ar

Página: <http://weblidi.info.unlp.edu.ar/catedras/adp>

Blog: <http://blogs.unlp.edu.ar/adp>

Entorno virtual: <http://ideas.info.unlp.edu.ar>



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Firmas del/los profesores responsables: